SIMULATED COMPUTER™

ATARI 400/800 VERSION

Award Winning Software from

EduSoft®

EduSoft WD-4 (disk), WT-4 (tape)
SIMULATED COMPUTER™

System Requirements

This program requires an Atari 400 or 800 with Atari BASIC. The disk version requires 24K of memory, and boots and runs automatically. The cassette includes two versions of the program: the first program on the tape requires 16K of memory, and does not include the "HELP" feature. The second program on the tape requires 24K of memory, and includes the "HELP" feature. Both programs are recorded, one after the other, on each side of the tape.

You are encouraged to make a backup copy of this package in case some disaster befalls the original. We remind you that it is illegal and, more importantly, immoral to give or sell copies of these programs to your friends or colleagues. (Theft is theft!)

Published by **EduSoft**, P.O. Box 2560, Berkeley, CA 94702.
© 1982 by Scott Steketee and James Wieder. Unauthorized duplication of disk, tape or documentation is illegal. Atari is a trademark of Atari, Inc.

SIMULATED COMPUTER MANUAL

CONTENTS

Introduction . . . . p. 2
Lesson 1: Getting Under Way . . . . . . p. 3
Lesson 2: Stepping Through . . . . p. 4
Lesson 3: Computer Algebra . . . . p. 6
Lesson 4: Decisions, Decisions . . . . . p. 7
Lesson 5: Countdown . . . . p. 9
Appendix A: Parts of a Computer . . . . p. 10
Appendix B: Commands . . . . p. 11
Appendix C: Computer Instructions . . . p. 12
Appendix D: Error Messages . . . . p. 13
Appendix E: Suggested Problems . . . . . p. 13
Appendix F: Sample Programs . . . . . . p. 14

Copyright 1980 by Steketee Educational Software
4639 Spruce St., Philadelphia, Pa. 19139

Additional copies of this manual may be purchased for $.50 each. For an order of 20 or more copies, the price is $.40 each. Include $1.00 shipping and handling with each order.

INTRODUCTION

The Simulated Computer demonstrates many of the basic principles of operation of a real computer. Like a real computer, it performs only one operation at a time, and works only with numbers. Its instructions are given in numeric form, and it can be programmed by providing the proper numeric codes to make it perform, one step after another, the steps of a complex task.

While it is based on the same principles, the operation of the Simulated Computer is in many ways easier to understand than the operation of a real computer.

1) The Simulated Computer uses ordinary decimal (base 10) numbers. Real computers operate in binary (base 2). (Programmers often use the related number systems octal [base 8] and hexadecimal [base 16] for their convenience.)
2) The Simulated Computer has 20 memory locations. Real computers have from about 4000 (4K) to about 64000 (64K) and even more.
3) The Simulated Computer uses only 10 types of instructions. A real computer might use 50, 100, or even more.
4) The Simulated Computer can execute about two instructions per second. A real computer such as the Apple II or TRS-80 can execute hundreds of thousands of instructions per second.

Learning to program the Simulated Computer will not teach you how to write machine language programs on a real computer, of course. But it will help you to understand and visualize the inner workings of a computer. Computers will no longer be quite the "mystery boxes" that they may appear to you now. And should you decide to learn machine language programming, or its cousin, assembly language programming, the understanding of the operating principles of a computer gained by working through this manual will prove to be very valuable indeed.

LESSON 1: GETTING UNDER WAY

1) First, load the Simulated Computer into your Apple or TRS-80. You can load it from tape or diskette, just as you would any other program.
2) Run the program, and look at the summaries it offers you. Don't worry about understanding the summaries yet.
3) When the computer has been drawn on the screen, look it over. Locate the basic parts of the computer--the input, output, memory, and CPU (central processing unit). The CPU contains the three registers (accumulator, program counter, and instruction register) shown on the screen, and it also contains all the electronics to perform the additions, subtractions, and other operations which the computer can perform.
4) Now load Sample Program 1 into the Simulated Computer. To begin, type the command "LOAD" and press return. Then type the 3-digit numbers shown in the right-hand column, and press return after each number. Finally, type "END" and press return.
5) Now run the program. Type the word "RUN" and press return. Watch it run, and see if you can tell what it is doing. The numbers will change too quickly for you to see each step, but you can at least get an idea of what the program does.
6) While the program is running, press the letter "B". This is used as the "break" key, and is used to stop a program that doesn't stop automatically. Now type the command "CONT" (for "continue") and press return.
7) Stop the program again using the "B". This time, type "HELP" and press return. When you're all done seeing the effect of this command, type "CONT" again.
8) Stop the program again. This time, type the command "NEW" and press return.

```
LOAD
00: 105
01: 306
02: 207
03: 807
04: 901
05: 000
06: 001
07: END
RUN
```

Sample Program 1

QUESTIONS ON LESSON 1

1) What did the program do?
2) What command was used to enter a program into the computer?
3) What command was used to make the computer execute the steps of the program, once the program was in the computer?
4) What key was used to stop the computer in the middle of running a program?

Explain in a complete sentence the effect of each of these commands:

5) CONT 6) HELP 7) NEW

LESSON 2: STEPPING THROUGH

```
LOAD
00: 105
01: 506
02: 207
03: 807
04: 901
05: 001
06: 002
07: END
RUN
```

Sample Program 2

1) Start the Simulated Computer, and LOAD Sample Program 2, using the same method you used in Lesson 1.
2) RUN the program to see what it does.
3) When the program stops, you will see an error message. Note the number of the error, and then type HELP to review the list of error messages and to find out what type of error it was.
4) When the Simulated Computer is on the screen again, type the command DISPON to turn the display on, and RUN the program again. Note the effect of the DISPON.
5) The program is running too fast for you to see the effect of each individual step. Type "B" to stop the program. Then start it again using the command RUNSTEP. This puts the computer in "single step" mode. Each time you press the space bar, the computer will either do <u>one</u> step, or display <u>one</u> comment.
6) Press the space bar. The number 00 should appear in the PC register. This indicates that the computer will start with the instruction in location 00.
7) FETCH: Press the space bar again. The comment which appears tells you what the computer is about to do. Press the space bar again. Notice which register was changed.
8) INCREMENT: Press the space bar to see the next comment. The computer is about to increment the Program Counter. Press the space bar again to see this happen.
9) EXECUTE: Look at the instruction in the instruction register (IR). The first digit tells the computer what operation to perform. (This digit is called the operation code, or op. code for short.) The last two digits tell the computer what memory location to use. (These two digits are called the address code.) Press the space bar to see the next comment. The first (1) tells the computer to load the accumulator (AC). The last two digits tell it to use the contents of location 05. Press the space bar to see this step happen.
10) Continue pressing the space bar to single step through the program. You will see the pattern FETCH an instruction, INCREMENT the Program Counter, and EXECUTE the instruction performed again and again. You should soon be able to predict what will happen at each step.

11) Remember that the first digit of the instruction in the IR is called the op code. As you single-step through the program, take note of the effect of each of the different op codes.
12) Now press "B" to stop the program. Type the command DISPOFF to turn off the display.
13) Type CONSTEP to continue execution of the program, still in single step mode. See if you can still predict the effect of each step, without the help of the comments.
14) Stop the program again, and turn the display on with DISPON. Now use the command RUNSLOW to run the program in "slow" mode. See if you can follow the computer's steps as it does them, at this slower pace.

QUESTIONS ON LESSON 2

1) What is the effect of Program 2?
2) What kind of error stopped the program? Why do you think this error occurred?
3) What does the command DISPON do?
4) What command turns off the display of comments?
5) When the computer begins a program, from what location does it take its first instruction?
6) The FETCH cycle changes the contents of which register?
7) The INCREMENT cycle changes the contents of which register?
8) The first digit of an instruction is called the ____ code.
9) The last two digits of an instruction are called the ____ code.
10) What command will continue the execution of a program, using slow mode?
11) What command will <u>start</u> the execution of a program, using single-step mode?
12) Complete this list of the effect of different op codes in Sample Program 2:

op code	effect
1	load the accumulator
5	____________________
2	____________________
8	____________________
9	____________________

13) For each computer instruction below, list the effect. You can use the HELP command to find the effects of op codes you don't know already--or look in the Appendix.

instruction	effect
109	load AC from loc 09
211	store ____ in loc ____
507	______ the AC by the contents of loc ____
804	______ from loc ____
913	______ to loc ____
315	______________
403	______________

14) Decode these additional instructions:

instruction	effect
612	______________
904	______________
715	______________
000	______________

15) a. Write an instruction to subtract the contents of location 12 from the contents of the accumulator.
    b. Write an instruction to jump to location 02.
    c. Write an instruction to output the contents of loc 19.

16) Refer to program #1 in lesson 1. Revise it so that it counts by twos instead of by ones. Test your revision by running it on the Simulated Computer.

17) Now write a program to start with 50 and count by tens. Check your program by running it.

LESSON 3: COMPUTER ALGEBRA

1) Start the Simulated Computer and LOAD Sample Program 3. Notice the effect of using the command LOAD-07 to load the program.
2) Now type RUN-07. Note the first value that appears in the Program Counter.
3) When the flashing # sign appears, type any number and press return. Each time the flashing # sign re-appears, type a number and press return. Repeat until you have typed a total of five or six numbers. Can you tell what the program does?

```
LOAD-07
07: 701
08: 101
09: 512
10: 820
11: 907
12: 007
13: END
RUN-07
```

Sample Program 3

4) Sample Program 3 accepts any number that you input, and multiplies it by 7. In the language of algebra, we could say that the computer "evaluates" the expression "$7x$" and prints the result. In the "equation" $y = 7x$, x stands for your input, and y stands for the computer's output.
5) Single-step through the program, using CONSTEP, so that you can see the steps that are involved. First comes the input instruction, then the calculation, then the output, and finally a jump instruction so that the program will repeat for a new value of x. You will be asked below to write some programs of your own, to do more complicated expressions. You can follow the same plan as in Sample Program 3, but the calculation section of your programs will be longer.

QUESTIONS AND PROBLEMS ON LESSON 3

1) Describe the difference between these commands: LOAD and LOAD-11.
2) When you give the command RUN-03, into what register will the number 03 be placed when the computer starts to run the program?
3) Write a program to evaluate y in the equation $y = x - 5$. Load your program starting at location 12, and run it to be sure it works. Write down both the program and the output.
4) Now do a program to evaluate $y = 5x - 8$. Run it for $x = 3$ and $x = 47$, and write down program and output.
5) Now write a program to evaluate $y = 3a + 5b$. Run it for $a=2$, $b=5$ and also for $a=-6$, $b=-1$, and check the results by hand to make sure your output is correct. (If you need a hint, look at Sample Program 6 in the Appendix.)
6) As a last challenge, write a program to evaluate $y = x^2 + 2x - 1$, and run it for $x=3$, $x=0$, and $x=-5$. (For a hint, look at Sample Program 7 in the Appendix.)

LESSON 4: DECISIONS, DECISIONS

1) Load Sample Program 4 on page 8, and run it for these values of input: -1, 3, -8, -6, and 2. Can you figure out what the program does?
2) To be useful in any real situation, a computer must be able to make decisions. This program makes a very simple decision: it decides whether a number is less than or equal to zero,

and outputs the number 1 if the answer is true, and outputs the number 0 if the answer is false. Notice how the skip instruction (004) is used to send the computer to one location (03) if the condition is true, and to another location (02 and then 05) if the condition is false.
3) Now write a program yourself to decide if the number input is less than zero. Make the computer print out 84 if the condition is true, and 70 if it is false. (The computer code for the letter T is 84; the code for F is 70.)
4) More complicated decisions can also be made. For instance, to decide if x is greater than 11, you could subtract 11 from x, and then decide if the result is greater than zero.

```
LOAD
00: 720
01: 004
02: 905
03: 807
04: 900
05: 808
06: 900
07:   1
08:   0
09: END
RUN

 Sample
Program 4
```

QUESTIONS AND PROBLEMS ON LESSON 4

1) Computer codes for the letters of the alphabet start with 65 for "A" and continue up through the alphabet. If you wanted the computer to output either "Y" or "N", what codes would you use for these two letters?
2) The first instruction of Sample Program 4 is 720, but the Simulated Computer does not contain a memory location 20. What is the meaning of an address code of 20, used in an input or output instruction?
3) Write a program to determine whether the value of x which is input is less than or equal to 7. Have the computer display the number 1 if the result is true, and the number 0 if it is false. Test the program using these values for x: x=6, x=7, and x=8. Write down your program and the output. (See Sample Program 8 if you need help.)
4) Now try a program to decide whether $x^2 - 3 = 2x$. Test it for x=2, x=3, and x=4. Write down the program and the output.

LESSON 5: COUNTDOWN

1) Load and run Sample Program 5. Examine it carefully to be sure you know how it works. Notice particularly the use of the subtraction instruction in location 03 and the skip instruction in location 04. These two instructions allow the computer to decide when to stop.
2) Modify the program so that it counts from 1 to 20.
3) Modify it to count from 10 to 20.
4) Modify it to count from 10 to 20 by 2's.

```
LOAD
00: 109
01: 212
02: 812
03: 411
04: 001
05: 000
06: 112
07: 310
08: 901
09:   1
10:   1
11:  10
12: END
RUN

 Sample
Program 5
```

PROBLEMS ON LESSON 5

1) Write a program that will count from 20 to 50 by fives.
2) Now write a program to count down from ten to zero. Careful--you'll have to change the type of decision your program makes.
3) Write a program which will print out the value of x^2 for all values of x from 10 to 20.
4) Finally, write a program to count down from 100 to 0 by fives, and to print the value of x and the value of $3x - 20$ for each time through the program.

APPENDIX A: PARTS OF A COMPUTER

When you start the Simulated Computer, you will see the screen divided into four parts, representing the four parts of a real computer. The four parts are described below.

Input Device:
This is the part of the computer which allows you to put information (programs or data) into the computer. For the Simulated Computer, this part includes the keyboard and the INPUT box on the screen.

Central Processing Unit:
This part, often called the CPU, includes the accumulator (AC), the program counter (PC), and the instruction register (IR). It also includes the necessary circuits to regulate the sequence of machine cycles, and the circuits to perform the various arithmetic operations. Inside the Apple II or TRS-80, all these functions are performed by a single chip called a micro-processor. The Apple uses a "6502" micro-processor; the TRS-80 uses a different type called a "Z-80" micro-processor.

Memory:
The twenty memory locations displayed on the screen are used to hold instructions or data. Each instruction is in the form of a number.

Output Device:
This is the part of the computer where the results of the program are printed or displayed. On the Simulated Computer, it is simply the output box on the screen.

The Central Processing Unit, or CPU, is the heart of the computer, and controls the operation of the rest of the machine through the Program Counter (PC) and Instruction Register (IR). Each computer instruction is executed in a three-step process, using three "machine cycles". You can see these machine cycles when you single-step through a program.

Machine Cycles:
1. Load the IR from the memory location determined by the PC.
2. Increment the PC.
3. Execute the instruction contained in the IR.

APPENDIX B: COMMANDS

NEW	Clear all registers and memory locations.
LOAD	Load a program or data into memory, starting at location 00.
LOAD-12	Load a program or data, starting at location 12. (Useful for correcting a mistake.)
	(Type "END" to stop loading.)
RUN	Run the program in memory, starting at location 00.
RUN-05	Run the program, starting at location 05.
RUNSLOW	Run the program slowly, with a delay before each machine cycle. To change the delay, see WAIT.
RUNSTEP	Run the program in single step mode. This will make the computer stop after each machine cycle, and wait for you to press the space bar.
CONT	Continue execution of the program. This command may be used to restart a program after a stop instruction (code 000), or after you stopped the program by pressing the "B" (break) key.
CONSLOW	Continue execution, but in slow mode.
CONSTEP	Continue in single-step mode.
WAIT-2	Change the delay used when the computer is in the RUNSLOW mode. WAIT-9 will give the slowest operation, and WAIT-1 will give the fastest.
DISPON	Turn on the display mode. In this mode, a message will appear at the top of the screen before each machine cycle, describing what the next machine cycle is about to do. In single step mode, you will press the space bar once to see the message, and then press it again to see the effect.
DISPOFF	Turn off the display mode.
HELP	Display on the screen a summary of commands, instructions, error codes, etc. The contents of all memory locations and registers will be saved, but the information in the input and output boxes will be lost.

As an additional feature, the letter "B" functions as a break key. If you press "B" at any time during execution of a program, the Simulated Computer will stop and wait for your next command. If you type CONT, CONSLOW, or CONSTEP, the computer will resume execution of the program in whichever mode you indicated.

APPENDIX C: COMPUTER INSTRUCTIONS

Each instruction is a three digit number. The first digit tells the computer what type of operation to perform, according to the list below. (This digit is called the op. code.) The final two digits tell the computer what memory location to use in performing the operation. (These digits are called the address code.) For instance, in the instruction 317, the op code is 3 and the address code is 17.

The address code must usually be between 00 and 19, or an error will occur. For input and output instructions (op codes 7 and 8), an address code of 20 is permitted, to indicate input directly to the accumulator or output directly from the accumulator. For stop or skip instructions (op code 0), the address code indicates under what circumstances a skip will take place.

Here is a list of the op codes:

0 Stop or skip the next instruction--
    000 stop
    001 skip if accumulator is negative
    002 skip if accumulator is positive
    003 skip if accumulator is zero
    004 skip if accumulator is negative or zero
    005 skip if accumulator is positive or zero
    006 skip if accumulator is not zero
1 Load the AC from the indicated memory location.
2 Deposit the contents of the AC into the indicated memory location.
3 Add the contents of the indicated memory location to the contents of the AC.
4 Subtract the contents of the indicated memory location from the contents of the AC.
5 Multiply the contents of the AC by the contents of the indicated memory location.
6 Divide the contents of the AC by the contents of the indicated memory location. (The result is truncated.)
7 Input a number to the indicated memory location. (An address code of 20 refers to the AC.)
8 Output the contents of the indicated memory location. (An address code of 20 refers to the AC.)
9 Jump to the instruction in the memory location indicated.

APPENDIX D: ERROR MESSAGES

Whenever an error occurs, the computer will output an error message. The number which accompanies the word "ERROR" gives some information about the condition which caused the error. Here is a list of the error codes:

1 Overflow. The result of a calculation was larger than 999 or smaller than -99.
2 Undefined value. An arithmetic operation or output was attempted, using an empty memory location or empty accumulator.
3 Invalid instruction. Either the op code is missing, or the address code refers to a non-existent location.
4 Can't continue. The computer can continue only if execution was halted by a stop instruction (000) or by pressing the "B" key (break key).
5 Invalid command. The only commands permitted are those listed in Appendix B.
6 Input error. This error is usually the result of typing letters when numeric input is required. (The # prompt indicates when numeric input is needed.)
7 Missing instruction. The memory location from which the computer is attempting to get its next instruction is empty.
8 Division by zero. This is not allowed.

APPENDIX E: SUGGESTED ADDITIONAL PROBLEMS

1) Write a program to input two numbers, add them, and output the result.
2) Write a program to evaluate $3x - 4$ for any value of x which is input.
3) Write a program to evaluate $3x^2 - 2x + 5$ for any value of x.
4) Write a program to evaluate x^y (x raised to the y power) for any values of x and y that are input.
5) Write a program to find the approximate square root of any number from 1 to 999, using the divide-and-average method.
6) Write a program to determine whether $x^2 - 2x - 15 \geq 0$ is true or false, for any value of x. The program should output a 0 if the statement is false, and a 1 if the statement is true.
7) Write a program to print the digits of the decimal expansion of a fraction. (For example, an input of 1 and 8 should give this output: 0 1 2 5.)

APPENDIX F: SAMPLE PROGRAMS

Sample
Program 6
(8a - 3b)

```
LOAD
00: 718  (a)
01: 719  (b)
02: 119  (b)
03: 510  (3b)
04: 217  (3b)
05: 118  (a)
06: 511  (8a)
07: 417  (-3b)
08: 820
09: 900
10:   3
11:   8
```

Sample
Program 7
(x^2-3x+5)

```
LOAD
00: 719  (x)
01: 119
02: 510  (3x)
03: 218
04: 119  (x)
05: 519  (x^2)
06: 418  (-3x)
07: 311  (+5)
08: 820
09: 900
10:   3
11:   5
```

Sample
Program 8
(Is $x < 17$?)

```
LOAD
00; 720  (x)
01: 408  (-17)
02: 001  (< 0?)
03: 906  (jump if false)
04: 809  (true: 1)
05: 900
06: 810  (false: 0)
07: 900
08:  17
09:   1
10:   0
```

Sample
Program 9
(Fibonacci)

```
00: 109
01: 308
02: 208
03: 808
04: 309
05: 209
06: 809
07: 901
08:   0
09:   1
```

Sample
Program 10
(Factorial)

```
00: 719
01: 119
02: 413
03: 218
04: 002
05: 910
06: 519
07: 219
08: 118
09: 902
10: 819
11: 000
12: 900
13:   1
```

Sample
Program 11
(Decimal Divide)

```
00: 719
01: 718
02: 119
03: 618
04: 217
05: 817
06: 518
07: 216
08: 119
09: 416
10: 006
11: 000
12: 515
13: 219
14: 903
15:  10
```

(Solution to
Problem 7,
Appendix E)

Changes and Improvements in the Atari Version

The Atari version of Simulated Computer contains several color, graphics, and control enhancements which are not available on other computers. The changes which affect the instructions in the Simulated Computer Manual are described below.

1. Break Key (See p. 3 of Manual): A program may be interrupted either by typing 'B' as described in the manual, or by holding down the trigger button on the joystick. If interrupted by using the joystick trigger, the program can be restarted by pressing the joystick in any direction.

2. Help Mode (See p. 3 of Manual): The HELP command is available on the disk version and on the 24K cassette version. It is not available on the 16K cassette version. The information presented by the HELP command is also contained in the Manual in Appendices B through D, pages 10 through 13. (The first program on each side of the cassette is the 16K version; the second program on each side is the 24K version.)

3. Single Step Mode (See p. 4 of Manual): When running the Simulated Computer in single step mode (RUNSTEP or CONSTEP), you may press the space bar or the joystick (any direction) to execute the next machine cycle.

4. Display Mode (See p. 4 of Manual): The Simulated Computer will automatically start up with Display Mode on. You may turn it off at any time by typing DISPOFF at the "?" prompt. (The status of the display mode is shown at the top of the screen.)

5. Speed of Operation: When running in other than step mode, you can control the speed of the Simulated Computer with the joystick. Press forward to speed it up; pull back to slow it down. When the program is running at maximum speed, the border around the Simulated Computer will change to bright yellow. For fastest operation, turn Display Mode off. If you don't have a joystick, you can still control the Simulated Computer's speed using the WAIT and RUNSLOW commands (see p. 5 of Manual).